İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Yıl: 4 Sayı: 8 Güz 2005/2 s.1-18

CURE, AGNES VE K-MEANS ALGORİTMALARINDAKİ KÜMELEME YETENEKLERİNİN KARŞILAŞTIRILMASI

Meral DEMİRALAY[*], A. Yılmaz ÇAMURCU[**]

ÖZET

Bu çalışmada, hiyerarşik kümeleme algoritmalarından CURE (Clustering Using REpresentatives) ve AGNES (AGglomerative NEsting) ile bölümleyici kümeleme algoritmalarından çok sık kullanılan k-means' in sentetik veri setlerinde uygulanmasıyla elde edilen sonuçların karşılaştırması açıklanmaktadır. Gerçekleştirilen uygulamalarda, k-means algoritmasının ayrık ve sıkışık bulutlar halindeki kümeleri başarıyla bulduğu görülmüştür. Bu algoritma benzer büyüklükteki küresel kümeleri bulabilirken, çok büyük kümeleri küresel de olsa parçalara ayırmaktadır. AGNES algoritması uygulamaları bu algoritmanın küresel kümeleri etkili bir şekilde bulduğunu, ancak sıradışı noktalara karşı çok duyarlı olduğunu göstermiştir. CURE algoritması uygulamalarında bu algoritmanın farklı büyüklüklerde ve farklı şekillerdeki kümeleri sıradışı noktalardan etkilenmeden başarıyla bulduğu görülmüştür. Ancak, CURE algoritmasıyla elde edilen kümelerin giriş parametrelerinin değerlerinden etkilendiği saptanmıştır.

Anahtar Kelimeler: Kümeleme, Hiyerarşik Kümeleme, K-Means, CURE, AGNES

COMPARISON OF CLUSTERING CHARACTERISTICS OF CURE, AGNES AND K-MEANS ALGORITHMS

ABSTRACT

In this study, applications on the synthetic datasets using hierarchical clustering algorithms, CURE (Clustering Using REpresentatives) and AGNES (AGglomerative NEsting), and a partitioning based clustering algorithm, k-means are compared. This applied study shows that k-means algorithm can find discrete and condensed clusters successfully. According to the results of k-means applications, this algorithm can be used to find similar sized and spherical clusters, but, it divides the big clusters into smaller partitions even they are spherical. Applications on AGNES algorithm show that AGNES can find spherical clusters effectively, but, it is very sensitive to the outliers. Applied studies on CURE algorithm show that this algorithm can find different sized and different shaped clusters effectively. On CURE applications, it is found out that, clustering process is not affected from outliers but it is very sensitive to the value of the input parameters.

Keywords: Clustering, Hierarchical Clustering, K-Means, CURE, AGNES

[*] *Marmara Üniversitesi, Fen Bilimleri Enstitüsü,Göztepe-İstanbul, meral.demiralay@gmail.com*
[**] *Marmara Üniversitesi, Teknik Eğitim Fakültesi, Göztepe-İstanbul, camurcu@marmara.edu.tr*

1. GİRİŞ

Kümeleme, en basit tanımıyla benzer özellik gösteren veri elemanlarının kendi aralarında gruplara ayrılmasıdır. Literatürde kümeleme analizini açıklayan bir çok tanım bulunmaktadır (Berkhin, 2002; Bilgin, 2003; Boutsinas ve Gnardellis, 2002; Han ve Kamber, 2001; Jain ve Dubes, 1988; Jain vd., 1999; Karypis vd., 1999; Mercer, 2003; Witten ve Frank, 1999). Bu tanımlara göre her küme temsil ettiği nesneleri en iyi şekilde ifade edecek şekilde düzenlenir. Kümeleme işleminin uygulandığı veri setindeki her bir veriye nesne adı verilir. Bu nesneler iki boyutlu düzlem üzerinde noktalarla gösterilir. Kümeleme analizi, veri indirgeme veya nesnelerin doğal sınıflarını bulma gibi çeşitli amaçlarla kullanılmaktadır (Bilgin, 2003; Han ve Kamber, 2001; Karypis vd., 1999). Kümeleme analizinin kullanıldığı sayısız uygulama alanı bulunmaktadır. Bu alanlardan en çok gündemde olanlar örüntü tanıma, veri analizi, resim tanıma, pazarlama, metin madenciliği, doküman toplama, istatistik araştırmaları, makine öğrenimi, şehir planlama, coğrafik analizler (deprem, meteoroloji, yerleşim alanları), uzaysal veritabanı uygulamaları, Web uygulamaları, müşteri ilişkileri yönetimi, sağlık ve biyoloji alanında yapılan araştırmalardır (Berkhin, 2002; Bilgin ve Çamurcu, 2003; Han vd., 2001; Jain vd., 1999).

Kümeleme analizini gerçekleştirmek için birçok kümeleme metodu geliştirilmiştir (Han ve Kamber, 2001; Jain ve Dubes, 1988; Mercer, 2003; Witten ve Frank, 1999). Bu çalışmanın konusu olan kümeleme metotlarından hiyerarşik kümeleme metodunda, her küme bir veri setindeki her bir nesnenin dizideki bir sonraki nesnenin içinde yer aldığı bir nesneler dizisidir (Guha vd., 2001). Bu dizinin en üst seviyesinde tüm nesneleri içeren tek bir küme ve en alt seviyesinde ise ayrı noktalardan oluşan tekil kümeler yer alır (Karypis vd., 1999; Xiong vd., 2004; Zhao ve Karypis 2002). Bu iki seviye arasında kalan her seviyedeki küme, bu küme ve bu kümenin bir alt (veya bir üst) seviyesindeki kümenin birleşimidir (veya ayrışımıdır) (Halkidi vd., 2001).

Bu çalışmada, hiyerarşik kümeleme algoritmalarından CURE (Clustering Using REpresentatives) ve AGNES (AGglomerative NEsting) ile bölümleyici kümeleme algoritmalarından k-means algoritmasının MATLAB programında ve sentetik veri setleri üzerinde gerçekleştirilen uygulamalarına ilişkin sonuçların karşılaştırması açıklanmaktadır.

Guha, Rastogi ve Shim tarafından ilk olarak SIGMOD 1998 konferansında sunulan CURE algoritması birleştirici bir kümeleme metodudur. Hiyerarşik metotların küresel olmayan ve farklı boyutlu kümeleri bulma konusundaki zayıflıklarını ve sıra

dışılıklara karşı hassasiyetlerini gidermek üzere ortaya konmuştur (Guha, 2000; Guha vd., 2001; Han ve Kamber, 2001).

AGNES (AGglomerative NEsting) algoritması, Kaufman ve Rousseeuw tarafından 1990 yılında sunulmuştur (Bilgin, 2003; Kaufman ve Rousseeuw, 1990). Aşağıdan yukarı doğru çalışan bir inşa yapısı izler. Başlangıçta her nesne ayrı bir küme olarak kabul edilir. Algoritmanın sonraki her adımında bu atomik kümelerden benzer özellik gösterenler birleştirilir. Herhangi bir sonlanma koşulu verilmezse kümeleme işlemi tamamlandığında bütün nesneler tek bir kümede toplanır (Anders, 2003; Han ve Kamber, 2001; Karypis vd., 1999; Szymkowiak vd., 2001; Witten ve Frank, 1999).

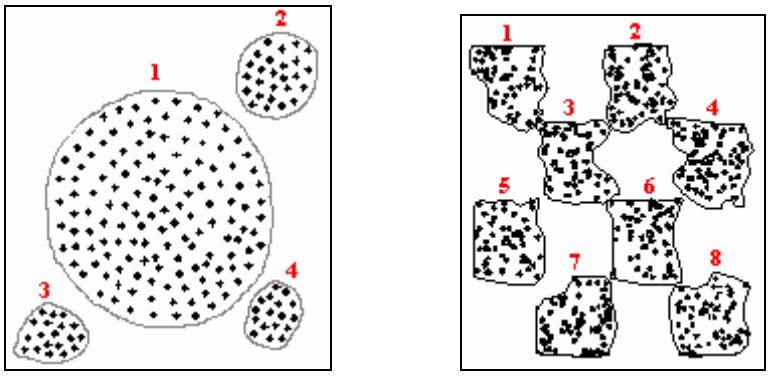
1967 yılında Mac Queen tarafından bulunan k-means algoritması, kümeleme problemini çözen en basit gözetimsiz öğrenme algoritmalarından biridir (MacQueen, 1967). Bölümleyici kümeleme tekniklerinden birisi olan k-means, bilimsel ve endüstriyel uygulamalarda en yaygın olarak kullanılan kümeleme algoritmaları arasında yer alır (Berkhin, 2002; Syed, 2004).

2. ALGORİTMALARIN VERİ SETLERİNE UYGULANMASI

2.1 Kullanılan Veri Setleri ve Yazılımlar

Hiyerarşik kümeleme metotları CURE, AGNES ve bölümleyici kümeleme metodu k-means'in sentetik veri tabanlarına uygulanarak karşılaştırılmasının açıklandığı bu çalışmanın iki temel amacı vardır. Bunlardan birisi, parametrelerin algoritmalardaki sonuçlara etkisinin görülmesi, diğeri ise farklı algoritmaların oluşturduğu kümelerin karşılaştırılarak yorumlanmasıdır.

Bu çalışmada, yuvarlaklar ve kareler görüntüsündeki iki ayrı sentetik veri seti kullanılmıştır. x ve y koordinatı bilgileri bulunan 200 adet noktayı içeren Yuvarlaklar veri seti birbirinden belirgin şekilde ayrılmış dört adet yuvarlaktan oluşmaktadır (Han, 2005). Kareler veri seti, sıkça kullanılan dama tahtası veri setinin bir versiyonudur (Ho ve KleinBerg, 1996; Ho ve KleinBerg, 2005). 16 kareli bir dama tahtasının ilgili sekiz adet siyah karesi rasgele seçilerek oluşturulmuş 486 adet siyah noktadan oluşmaktadır. Bu noktalar doğal olarak sekiz kümede toplanmaktadır. Yuvarlaklar ve Kareler veri setlerinin iki boyutlu düzlem üzerindeki görüntüleri Şekil 1 de görülmektedir.



a b

Şekil 1. Veri Setlerinin 2 Boyutlu Düzlemdeki Görüntüleri ve Küme Numaraları; a. Yuvarlaklar Veri Seti, b. Kareler Veri Seti.

Guha ve diğerleri (2001) tarafından verilen sözde kodlar ve Vipin Kumar tarafından gönderilen C programlama dilinde hazırlanmış temel kodlardan yararlanılarak MATLAB'de CURE algoritmasının yazılımı yeniden oluşturulmuştur. K-means algoritmasının programa uygulanmasında Roger Jang tarafından oluşturulan k-means fonksiyonu (Jang, 2005) ve Jon Shlens tarafından oluşturulan clusterdemo programı (Shlens, 2002) örnek alınmıştır. AGNES algoritması, MATLAB programında hazır olarak bulunan cluster fonksiyonu kullanılarak gerçekleştirilmiştir.

2.2 K-Means Algoritması ve Uygulaması

En eski kümeleme metotlarından biri olan k-means algoritmasının genel mantığı n adet veri nesnesinden oluşan bir veri setini, giriş parametresi olarak verilen k adet kümeye bölümlemektir. Amaç, gerçekleştirilen bölümleme işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin minimum olmasını sağlamaktır. Küme benzerliği, kümenin ağırlık merkezi olarak kabul edilen bir nesne ile kümedeki diğer nesneler arasındaki uzaklıkların ortalama değeri ile ölçülmektedir (Han ve Kamber, 2001; Berkhin, 2002).

K-means algoritmasının işlem basamakları şöyledir:

1. Adım: İlk küme merkezleri belirlenir. Bunun için iki farklı yol vardır. Birinci yol nesneler arasından küme sayısı olan k adet rasgele nokta seçilmesidir. İkinci yol ise merkez noktaların tüm nesnelerin ortalaması alınarak belirlenmesidir,

2. Adım: Her nesnenin seçilen merkez noktalara olan uzaklığı hesaplanır. Elde edilen sonuçlara göre tüm nesneler k adet kümeden kendilerine en yakın olan kümeye yerleştirilir,
3. Adım: Oluşan kümelerin yeni merkez noktaları o kümedeki tüm nesnelerin ortalama değeri ile değiştirilir,
4. Adım: Merkez noktalar değişmeyene kadar 2. ve 3. adımlar tekrarlanır.

K-means algoritmasında her bir nesnenin merkez noktalara uzaklığını hesaplamak için kullanılan dört farklı formül aşağıda açıklanmaktadır (MacQueen, 1967; Mercer, 2003):

Öklit Uzaklığı - Öklit Uzaklığının Karesi (Euclidean Distance - Squared Euclidean Distance): Öklit uzaklığı ve Öklit uzaklığının karesi formülleri ile standartlaştırılmış verilerle değil, işlenmemiş verilerle hesaplama yapılır. Öklit uzaklıkları kümeleme analizine sıradışı olabilecek yeni nesnelerin eklenmesinden etkilenmezler. Ancak boyutlar arasındaki ölçek farklılıkları öklit uzaklıklarını önemli ölçüde etkilemektedir. Öklit uzaklık formülü en yaygın olarak kullanılan uzaklık hesaplama formülüdür. Öklit ve Öklit uzaklığının karesinin formülleri aşağıda görülmektedir.

Öklit uzaklık formülü : $$distance(x, y) = \{\sum_i (x_i - y_i)^2\}^{1/2} \quad (1)$$

Öklit uzaklığının karesi formülü : $$distance(x, y) = \sum_i (x_i - y_i)^2 \quad (2)$$

City-block (Manhattan) Uzaklık Formülü (City-block (Manhattan) Distance): Manhattan uzaklığı boyutlar arasındaki ortalama farka eşittir. Bu ölçüt kullanıldığında farkın karesi alınmadığı için sıradışılıkların etkisi azalır. Manhattan uzaklığının formülü aşağıda görülmektedir.

Manhattan uzaklık formülü: $$distance(x, y) = \sum_i |x_i - y_i| \quad (3)$$

Chebychev Uzaklığı (Chebychev Distance): Chebychev uzaklığı iki nesne arasındaki mutlak maksimum uzaklığa eşittir. Chebychev uzaklığının formülü aşağıda görülmektedir.

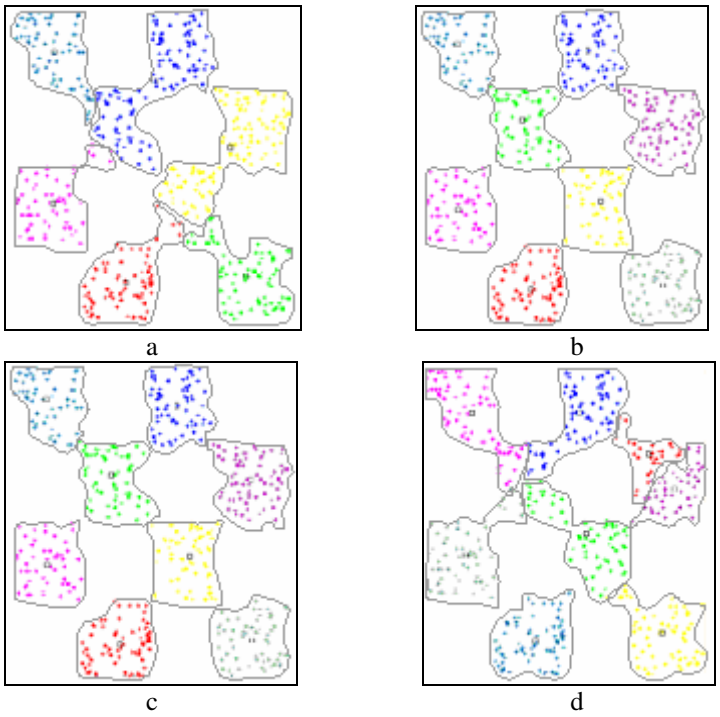
Chebychev uzaklık formülü: $$distance(x, y) = \max |x_i - y_i| \quad (4)$$

distance(x,y) : x ve y noktaları arasındaki uzaklık, hata parametresidir.
x ,y : aralarındaki uzaklık hesaplanan nesneleri uzayda temsil eden noktalardır.

K-means algoritmasının en büyük eksikliği k değerini tespit edememesidir. Bu nedenle başarılı bir kümeleme elde etmek için farklı k değerleri için deneme-yanılma yönteminin uygulanması gerekmektedir. Küme sayısının ve uzaklık

formüllerinin etkisini görmek için yuvarlaklar ve kareler setinde yapılan uygulamalar aşağıda açıklanmaktadır.

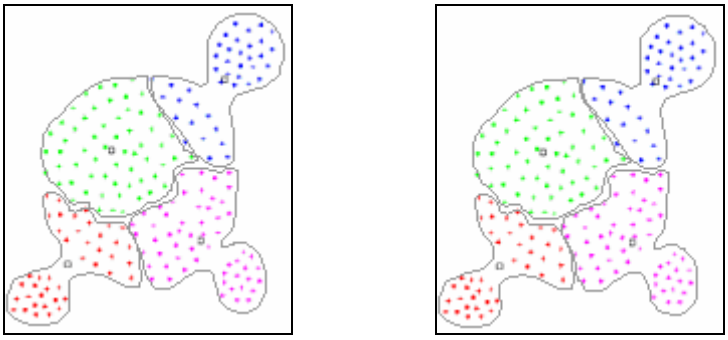
Yukarıda açıklanan k-means algoritmasına ilişkin formüllerin etkisini incelemek amacıyla kareler veri setlerinde küme sayısı k=6 alınarak Öklit uzaklığı formülü ile ve k=8 seçilerek Öklit uzaklığı, Öklit uzaklığının karesi ve Chebychev uzaklığı formülleri ile gerçekleştirilen uygulamalar sonucu elde edilen kümeler Şekil 2'de görülmektedir. Öklit uzaklığı formülünde k=6 alındığında anlamlı kümeler elde edilememektedir. Öklit uzaklığı ve Öklit uzaklığının karesi formüllerinin her ikisinde de k=8 seçildiğinde veri setindeki ideal kümelenmeler aynı şekilde bulunmuştur. Chebychev formülü ise veri setinin ideal kümelenmesinden uzak bir sonuç vermektedir. Bu metotta iki nesne arasındaki maksimum uzaklık değeri dikkate alındığından nesneler arasındaki komşuluklar farklı olarak bulunmuştur.



Şekil 2. K -Means Algoritmasının Kareler Veri Setlerinde Küme Sayısı k=6 İçin a. Öklit Uzaklığı Formülü ile, k=8 Alındığında, b. Öklit Uzaklığı, c. Öklit Uzaklığının Karesi, d. Chebychev Uzaklığı Formülleri İle Elde Edilen Kümeler.

Şekil 3'te görüldüğü gibi k-means algoritması, yuvarlaklar veri setine k=4 alınarak Öklit uzaklığı ve Öklit uzaklığının karesi formülleri ile uygulandığında başarılı olamamıştır. Büyük küme, küçük olan üç küme tarafından paylaşılmaktadır. K-means algoritması, hata parametresinin değerini minimum yapmak için büyük kümeleri bölerek mümkün olduğunca birbirinden ayrık ve kendi içinde sıkışık kümeler bulmaya çalışmaktadır.

Buna göre, k-means algoritmasının küresel kümelerde, her zaman doğru kümeleri bulamadığı ancak küme sayısı doğru seçildiğinde ayrık ve sıkışık bulutlar şeklindeki kümeleri etkili bir şekilde bulabildiği söylenebilir.



a b

Şekil 3. Yuvarlaklar Veri Setinde K-Means Algoritması Kullanılarak k=4 İçin a. Öklit Uzaklığı; b. Öklit Uzaklığının Karesi Formülleri İle Elde Edilen Sonuçlar.

2.3 CURE Algoritması ve Uygulaması

CURE algoritması, her kümenin sabit sayıda örneklem nokta ile temsil edildiği ve her adımda istenen küme sayısı elde edilene kadar örneklem noktaları en yakın olan kümelerin birleştirildiği aşağıdan yukarıya doğru çalışan hiyerarşik bir kümeleme algoritmasıdır. Her adımda yeni oluşturulan kümelerin örneklem noktalarını bulmak için birleşen kümelerin örneklem noktaları bir daraltma katsayısı ile çarpılır. Bu durumda algoritmanın doğru kümelenmeleri bulması üç parametrenin değerine bağlıdır: küme sayısı (k), örneklem nokta sayısı (rep_say), ve daraltma katsayısı (α). CURE algoritmasının çalışmasındaki işlem basamakları aşağıda görülmektedir (Guha, 2000; Guha vd., 2001):

1. Her küme için sabit sayıda ve küme içinde dağınık olarak yerleşmiş c adet örneklem nokta seçilir,

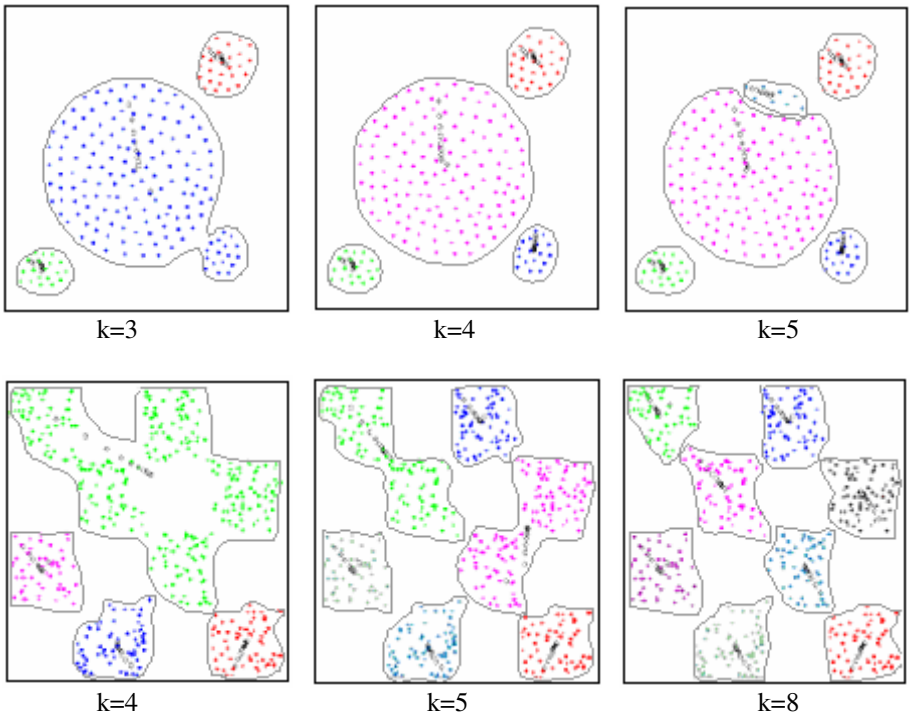
2. İki küme arasındaki uzaklık, bu kümelere ait örneklem noktalar arasındaki Öklit uzaklığı hesaplanarak elde edilir,
3. En yakın küme çifti birleştirilir,
4. Oluşan yeni kümenin örneklem noktaları bulunur. Bu işlem için yeni kümenin alt kümelerinden merkeze en yakın olan c adet nokta seçilir. Bu noktalar daraltma katsayısı α ile çarpılarak merkeze doğru yaklaştırılır,
5. Küme sayısı, kümeleme algoritmasında giriş parametresi olarak verilen k değerine ulaşana kadar 2, 3 ve 4. adımlar tekrarlanır.

Burada, CURE algoritmasının doğru kümeleri bulmasındaki, küme sayısı (k), örneklem nokta sayısı (c) ve daraltma katsayısı (α) gibi üç önemli faktörün, MATLAB'da farklı veri setlerindeki etkilerini görmek için deneysel çalışmalar yapılmıştır.

Küme Sayısının (k) Algoritmadaki Etkisinin İncelenmesi : CURE algoritmasında küme sayısı (k) kümeleme işleminin sonlanma koşulunu oluşturmaktadır. k değerinin kümelenme üzerindeki etkisini görmek için CURE algoritması yuvarlaklar veri setinde k=3, 4, 5 alınarak ve kareler veri setinde k=4, 6, 8 değerleri ile uygulanmıştır. k değerinin algoritmadaki etkisinin açık olarak görülebilmesi için örneklem nokta sayısı ve daraltma katsayısı değerleri sabit tutulmuştur. Kümelerin örneklem noktaları, yuvarlakların içindeki küçük karelerle gösterilmektedir.

Şekil 4'de CURE algoritmasının yuvarlaklar veri seti üzerinde, daraltma katsayısı α=0.2 ve örneklem nokta sayısı c=10 değerleri sabit tutularak, farklı k değerleri için uygulanması görülmektedir. Küme sayısı k=3 için elde edilen kümelenmede 2. ve 4. yuvarlaklar birleşmekte, k=4 seçildiğinde ideal kümelenme elde edilmekte ve k=5 alındığında veri setindeki en büyük yuvarlak bölünerek yeni bir küme oluşmaktadır. Görüldüğü gibi CURE algoritmasında yuvarlak kümelerin başarıyla bulunması için örneklem nokta sayısı ve daraltma katsayısı parametrelerinin farklı değerlerle denenmesi gerekmektedir.

Kareler veri setinde farklı k değerleri için α=0.3 ve c=10 alınarak elde edilen sonuçlar Şekil 3'de görülmektedir. k=4 seçildiğinde 1, 2, 3, 4 ve 6. kareler birleşerek 1. kümeye, 5, 7, ve 8. kareler ise doğru şekilde kümelenerek 2., 3. ve 4. kümelere yerleşmişlerdir. k=6 değeri için elde edilen kümelenmede, 1. ve 3. kareler birleşerek 1. kümeyi, 4. ve 6. kareler birleşerek 3. kümeyi oluşturmuştur. Diğer kümeler yaklaşık olarak doğru olarak bulunmuştur. Veri setindeki en anlamlı kümelenme k=8 alınarak bulunmuştur. CURE algoritması kare şeklindeki biçimleri de başarıyla kümelendirebilmektedir.

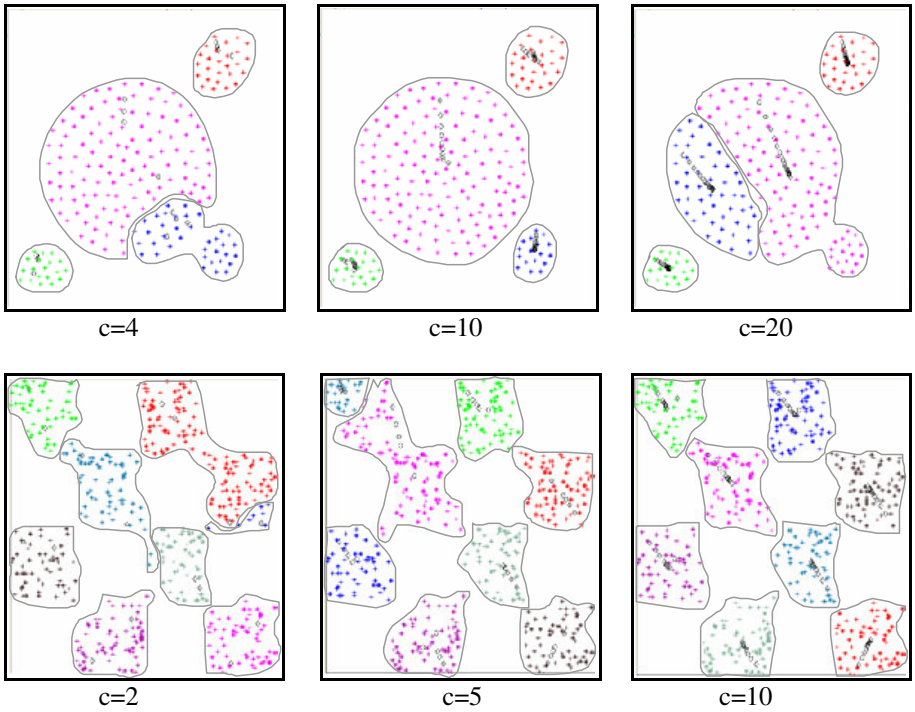


Şekil 4. CURE Algoritmasında, Yuvarlaklar Veri Setinde Daraltma Katsayısı α=0.2, Örneklem Nokta Sayısı c= 10 Alınarak k=3, 4, 5 Seçilince Ve Kareler Veri Setinde Daraltma Katsayısı α=0.3, Örneklem Nokta Sayısı c=10 Alınarak k=4, 6, 8 Belirlendiğinde Oluşan Kümeler.

Örneklem Nokta Sayısının (c) Algoritmadaki Etkisinin İncelenmesi : Örneklem nokta sayısı (c), bir kümeyi temsil etmek için kullanılan nesne sayısıdır. Şekil 5'de yuvarlaklar veri setinde α=0.2 ve k=4 alınarak c=4, 10, 20 değerleri için oluşan küme sonuçları görülmektedir. c=4 küçük bir değer olduğundan küçük ve ayrık kümelerin geometrik şekillerini bulabilmekte, ancak büyük yuvarlağı iki kümeye bölmektedir. c=10 değeri için kareler veri setinde de olduğu gibi başarılı bir kümeleme gerçekleşmektedir. c=20 için, örneklem noktaların kendi aralarında uzun bir küme oluşturarak uzamış kümeler bulma eğiliminde oldukları görülmektedir. Örneklem nokta sayısı yüksek bir değer aldığında, CURE algoritması tüm-noktalar yaklaşımlı algoritmalara benzer davranış gösterir ve uzatılmış biçimde görülen

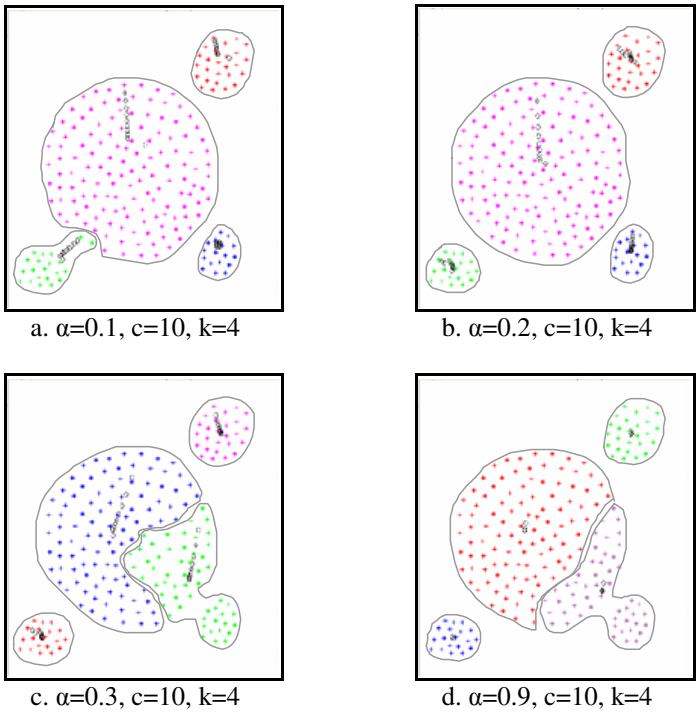
kümeler bulmaya çalışır. Bu durum, algoritmanın yavaş çalışmasına ve aralarında ortak komşu noktalar bulunan ayrık veri gruplarının birleşmesine neden olmaktadır.

Şekil 5'de kareler veri setinde daraltma katsayısı α=0.3 ve küme sayısı k=8 alındığında, örneklem sayısı c=2, 5, 10 değerleri için CURE algoritmasının kümeleme sonuçları görülmektedir. c=2 olarak seçildiğinde 2. ve 4. kareler birleşmiş bir küme ve 4 no.lu kareye ait olan bazı noktalar da yeni bir küme oluşturmuş, 3 nolu kareye ait bazı noktalar da 1 ve 6 no.lu karelerin kümelerine girmiştir. c=5 seçildiğinde 1 numaralı küme ikiye bölünmüştür. Örneklem nokta sayısı 10 değerini aldığında ise, kümeler iyi bir şekilde oluşmaktadır. Örneklem nokta sayısı küçük seçildiğinde, algoritma merkez nokta tabanlı algoritmalar gibi çalışmakta ve başarılı sonuç üretememektedir. Bu nedenle de geometrik şekiller bulunamamaktadır.



Şekil 5. CURE Algoritmasında, Yuvarlaklar Veri Setinde Daraltma Katsayısı α=0.2, Küme Sayısı k=4 Alınarak Örneklem Sayısı c=4, 10, 20 İçin Ve Kareler Veri Setinde α=0.3, k=8 Alındığında c=2, 5, 10 İçin Elde Edilen Kümeler.

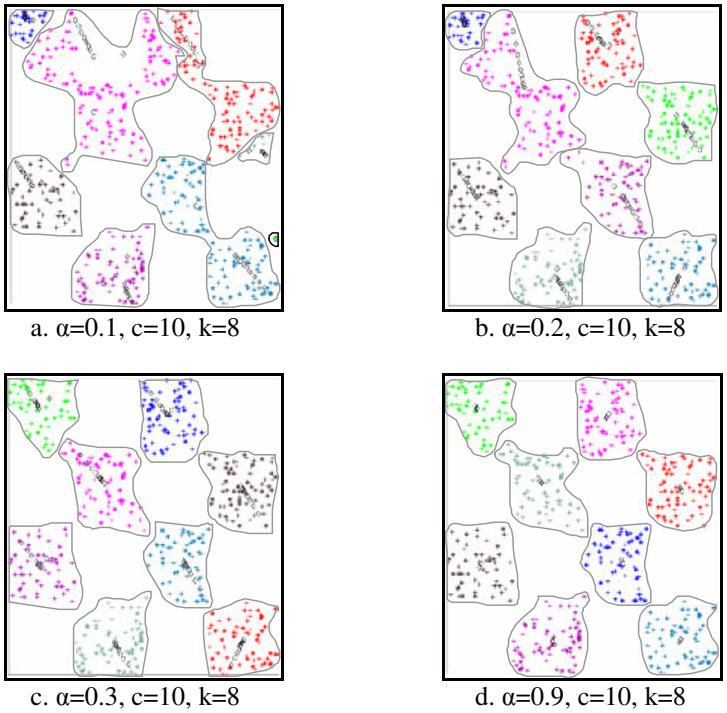
Daraltma Katsayısının (α) Algoritmadaki Etkisi : CURE algoritmasında α'nın kümeleme işlemi üzerindeki etkisini incelemek için yuvarlaklar ve kareler veri setlerinde gerçekleştirilen uygulamaların sonuçları Şekil 6 ve Şekil 7'de görülmektedir. Şekil 6a'da görüldüğü gibi, α=0.1 seçildiğinde 1. kümenin örneklem noktaları yeteri kadar merkeze yaklaşamadığı için 2. kümenin noktaları da 1. kümeye kaymaktadır. Şekil 7a'da α=0.1 seçildiğinde örneklem noktalar merkezden uzak kaldığından kümelenme doğru bir şekilde gerçekleşmemektedir. Sonuçta α=0.1 seçildiğinde dağınık (scattered) noktalar çok az kaydırıldığından doğru kümeler bulunamamaktadır. α katsayısı çok küçük seçilirse CURE algoritması sıradışı noktalara karşı daha duyarlı olmaktadır. Yuvarlaklar veri setinde α=0.2 alındığında, kareler veri setinde ise α=0.3 seçildiğinde doğru kümeler elde edilmektedir. α=0.9 seçildiğinde Şekil 6d ve Şekil 7d'de elde edilen sonuçlara göre örneklem noktaların kümenin merkez noktasında toplanarak algoritmanın merkez tabanlı kümeleme algoritmaları gibi çalışmasına neden olmaktadır.



a. α=0.1, c=10, k=4 b. α=0.2, c=10, k=4

c. α=0.3, c=10, k=4 d. α=0.9, c=10, k=4

Şekil 6. Yuvarlaklar Veri Setinde Farklı α Değerleriyle Gerçekleştirilen CURE Algoritması Uygulamalarının Sonuçları



a. α=0.1, c=10, k=8 b. α=0.2, c=10, k=8

c. α=0.3, c=10, k=8 d. α=0.9, c=10, k=8

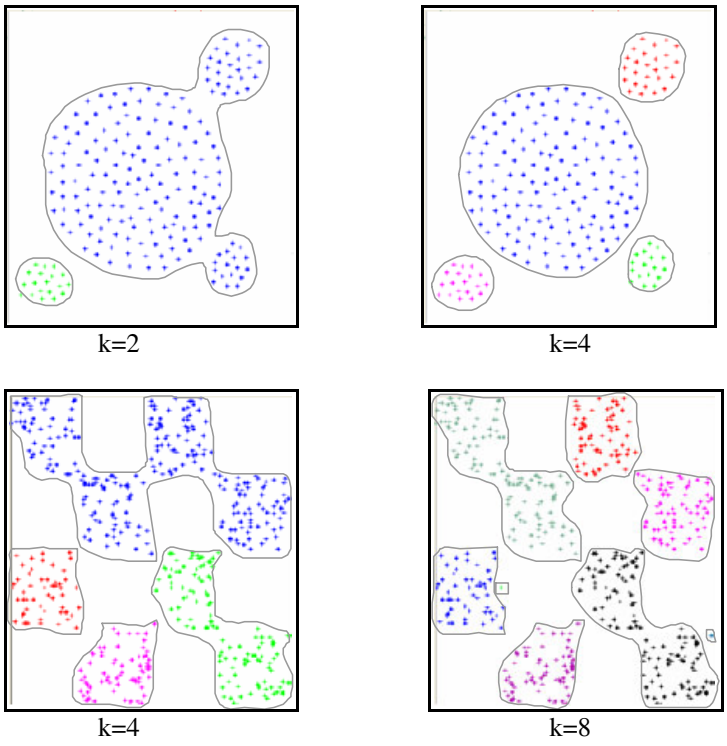
Şekil 7. Kareler Veri Setinde Farklı α Değerleriyle Gerçekleştirilen CURE Algoritması Uygulamalarının Sonuçları.

2.4 AGNES Algoritması ve Uygulaması

AGNES algoritması aşağıdan yukarı doğru çalışan bir inşa yapısı izler. Başlangıçta her nesne ayrı bir küme olarak kabul edilir. Algoritmanın sonraki her adımında bu atomik kümelerden benzer özellik gösterenler birleştirilir. Her birleştirme işleminden sonra toplam küme sayısı bir azalır. İstenen sayıda küme elde edildiğinde veya en yakın iki küme arasındaki uzaklık verilen eşik değere ulaştığında birleştirme işlemi sona erer. Herhangi bir sonlanma koşulu verilmezse kümeleme işlemi tamamlandığında bütün nesneler tek bir kümede toplanır (Anders, 2003; Han ve Kamber, 2001; Karypis vd., 1999; Szymkowiak vd., 2001; Witten ve Frank, 1999).

Küme Sayısının (k) Algoritmadaki Etkisi : Küme sayısı parametresi, k, AGNES algoritmasının sonlanma koşulunu belirler. İstenen küme sayısı elde edildiğinde

kümeleme işlemi sona erer. Kareler ve yuvarlaklar veri setlerinde farklı k değerleri için AGNES algoritması ile elde edilen sonuçlar Şekil 8'de görülmektedir. AGNES algoritması belirgin küresel kümelerden oluşan yuvarlaklar veri setine doğru küme sayısı verilerek uygulandığında başarılı sonuçlar vermektedir. Kareler veri setinde, k=8 olarak uygulandığında farklı karelerin ayrılması gerekirken birleşerek kümelendikleri ve ayrıca kare içindeki bazı noktaların kendi başına bir küme oluşturdukları görülmektedir. Bunun nedeni, AGNES algoritmasının sıradışı noktalardan çok fazla etkilenmesidir.



Şekil 8. AGNES Algoritmasının Yuvarlaklar Ve Kareler Veri Setlerinde Oluşturduğu Kümeler.

3. SONUÇLAR

Çalışmada, k-means, CURE ve AGNES algoritmalarının mevcut sentetik veri setleri üzerinde uygulanması ile elde edilen sonuçlara göre;

- CURE ve AGNES algoritmaları küresel kümeleri bulma konusunda k-means algoritmasından çok daha başarılıdır.
- K-means algoritması sıkışık bulutlar halindeki kümeleri bulurken CURE algoritması ile benzer sonuçlar elde etmiştir.
- K-means algoritması genel olarak küresel kümeleri bulabilmektedir. Ancak k-means algoritması büyük boyutlu küresel kümelerin bulunmasında başarısız olmuştur. K-means algoritmasında bu tip kümeler hatanın karesi değerini (hata parametresi) azaltmak için bölümlere ayrılmıştır.
- K-means algoritması ile elde edilen sonuçlar, seçilen hata parametresi hesaplama formülüne bağlı olarak değişmektedir.
- CURE algoritmasının başarılı sonuçlara ulaşabilmesi, parametrelerinin doğru seçilmesine bağlıdır. Bu nedenle en iyi sonucun bulunması için algoritmanın aynı veri seti üzerinde birkaç kez tekrarlanması gerekebilir.
- CURE algoritması, küresel ve şekilsiz kümelerin bulunmasında oldukça başarılı sonuçlar üretmektedir.
- AGNES algoritması, küresel olmayan kümelerde kötü sonuçlar vermektedir.

4. DEĞERLENDİRME

Literatürde, kümeleme algoritmalarının performans ve kullanışlılık açısından karşılaştırmasına yer veren pek çok çalışma vardır. Bu çalışmaların bir kısmında yeni bir metodu tanıtmak için diğer algoritmalarla uygulamalı karşılaştırma yapılırken, bir kısmında da hiçbir uygulama yapılmadan bilgi verme amaçlı olarak birden fazla kümeleme algoritmasına yer verilmektedir.

Bu çalışmada, incelenen kümeleme algoritmalarında küme oluşumlarının işlem basamakları bilgisayar ekranında izlenebilmektedir. Burada, karşılaştırılan kümeleme algoritmalarının her birinin parametrelere karşı duyarlılığı ve farklı veri setleri üzerindeki davranışları görülebilmektedir. Çalışmada, algoritmaların önceden bulunmuş sonuçlar kullanılarak değil, değişik veri setlerinde uygulanması ile elde edilen bilgilerle karşılaştırması yapılmaktadır.

Bu çalışmada, gerçekleştirilen uygulamalar ve elde edilen bulgular CURE algoritmasının AGNES ve k-means algoritmalarından daha güçlü olduğunu göstermektedir. CURE, AGNES ve k-means algoritmalarının farklı veri setleri

üzerinde çeşitli parametre değerleri denenerek yapılan uygulamalar sonucunda, CURE ve AGNES algoritmalarının küresel kümeleri rahatlıkla bulduğunu ancak k-means algoritmasının küçük kümelerin bulunduğu veri setlerinde büyük küresel kümeleri böldüğünü göstermektedir. Mercer ve Guha çalışmalarında (Guha vd., 2001; Mercer, 2003) k-means algoritmasının sadece eşit büyüklüklerdeki küresel kümeleri bulduğunu açıklamaktadır. Başka bir çalışmada ise k-means algoritmasının farklı boyutlardaki kümeleri her zaman ayırt edemediği belirtilmektedir (Wang ve Zaiane, 2002). Ancak bazı çalışmalarda, k-means algoritmasının bu özelliği vurgulanmayarak sadece küresel kümeleri bulabildiği belirtilmektedir (Fasulo, 1999; Guha vd., 2001). K-means algoritmasının sıradışı noktalardan etkilendiği hem elde edilen bulgularda, hem de yapılan çalışmalarda görülmektedir (Fasulo, 1999). CURE algoritması şekilsiz ve farklı büyüklüklerdeki kümeleri bulabilmektedir. Gerçekleştirilen uygulamalarda CURE algoritmasında hesaplamalar için çok uzun zaman harcandığı görülmektedir. Bu algoritmanın performansının ayrıntılı olarak incelenebilmesi için güçlü bilgisayar donanımları gerekmektedir. Bu çalışmada karşılaşılan bu sorun, Valgeirsson, Erlingsson, Einarson tarafından gerçekleştirilen çalışmada da belirtilmektedir (Valgeirsson vd., 2003). Baltrunas ve Gordevicius çalışmalarında CURE algoritmasının hesaplanabilir karmaşıklığına bir çözüm getirmektedirler (Baltrunas ve Gordevicius, 2005). Güçlü bilgisayar donanımları kullanılarak yapılan uygulamalarda ise CURE algoritmasının performansının iyi olduğu açıklanmaktadır (Guha vd., 2001). Bu makaledeki deneysel çalışmalarda güçlü bilgisayar donanımının sonuçlara çabuk ulaşmada çok önemli olduğu da görülmüştür.

5. KAYNAKÇA

Anders K-H., (2003), A Hierarchical Graph Clustering Approach to Find Groups of Data, Institute of Cartography and Geoinformatics University of Hannover.

Baltrunas L. ve Gordevicius J., "Implementation of CURE Clustering Algorithm", Technical Report, http://www.inf.unibz.it/dis/teaching/DWDM05/reports/cure.pdf ; Erişim tarihi: 17/04/2005.

Berkhin P., (2002), Survey of Clustering Data Mining Techniques, San Jose, California, USA, Accrue Software Inc..

Bilgin T., (2003), Veri Madenciliğinde Kümeleme Analizi Yöntemi Uygulaması, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Kontrol Eğitimi.

Bilgin T. ve Çamurcu Y., (2003), "A Data Mining Application on Air temperature Database", Lecture Notes in Computer Science, Springer-Verlag.

Boutsinas B. ve Gnardellis T., (2002), "On Distributing the Clustering Process", Pattern Recognition Letters 23, 999-1008.

Fasulo D., (1999), An Analysis of Recent Work on Clustering Algorithms, Technical Report, 01-03-02, Department of Computer Science & Engineering, University of Washington.

Guha S., (2000), Approximation Algorithms for Facility Location Problems, Stanford University Computer Science.

Guha S., Rastogi R. ve Shim K., (2002), "CURE: An Efficient Clustering Algorithm for Large Databases", Information Systems 26, 1, 35-58.

Halkidi M., Batistakis Y. ve Vazirgiannis M., (2001), On Clustering Validation Techniques, Kluwer Academic Publishers.

Han E.-H., (2005), İnternette Kişisel Görüşme, Research Associate, Department of Computer Science, University of Minnesota, Minneapolis.

Han J. ve Kamber M., (2001), Data Mining Concepts and Techniques, Morgan Kauffmann Publishers Inc.

Han J., Kamber M. ve Tung A. K. H., (2001) "Spatial Clustering Methods in Data Mining: A Survey", Geographic Data Mining and Knowledge Discovery, H. Miller ve J. Han (ed.), Taylor and Francis.

Ho T. K. ve KleinBerg E. M., (1996), "Building Projectable Classifiers of Arbitrary Complexity", Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 880-885.

Ho T. K. ve KleinBerg E. M., Checkboard Dataset
http://www.cs.wisc.edu/math-prog/mpml.html ; Erişim tarihi: 07/01/2005.

Jain A. K. ve Dubes R. C., (1988), "Algorithms for Clustering Data", Englewood Cliffs, New Jersey, 07632, Prentice Hall.

Jain A. K., Murty M. N. ve Flynn P. J., (1999), "Data Clustering: A Review", ACM Computing Surveys, 31, 3.

Jang R., Computer Science Department of Tsing Hua University, Taiwan, http://neural.cs.nthu.edu.tw/jang/matlab/demo/ ; Erişim tarihi: 06/06/2005.

Karypis G., Han E. H. ve Kumar V., (1999), "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling", IEEE Computer 32, 8, 68-75.

Kaufman L. ve Rousseeuw P. J., (1990), Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley and Sons.

MacQueen J., (1967), Some Methods for Classification and Analysis of Multivariate Observations, Berkeley, University of California Press.

Mercer D. P., (2003), "Clustering Large Datasets", http://www.stats.ox.ac.uk/~mercer/documents/transfer.pdf ; Erişim tarihi: 13/05/2005.

Shlens J., e-posta: jonshlens@ucsd.edu , Erişim tarihi: 09/10/2002.

Syed A. A., (2004), Performance Analysis of K-Means Algorithm and Kohonen Networks, Yüksek Lisans Tezi, Florida Atlantic University, Master of Science .

Szymkowiak A., Larsen J. ve Hansen L. K., (2001), "Hierarchical Clustering for Data Mining", KES'2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, Osaka-Japan.

Valgeirsson A. G., Erlingsson B. ve Einarson I. S., (2003), Using Clustering to Index Image Descriptors: A Performance Evaluation, ReykJavik University, B.Sc. Project.

Wang W. ve Zaiane O. R., (2002), "Clustering Web Sessions by Sequence Alignment", SIGMOD Conference.

Witten I. H., Frank E., (1999), "Data Mining: Practical machine learning tools with Java implementations.", San Francisco, Morgan Kaufmann.

Xiong H., Steinbach M., Tan P.-N. ve Kumar V., (2004), "HICAP: Hierarchial Clustering with Pattern Preservation", In Proc. of the Fourth SIAM International Conf. on Data Mining (SDM'04), Florida, USA.

Zhao Y. ve Karypis G., (2002), ”Clustering in Life Sciences.”, Technical Report, Department of Computer Science and Engineering University of Minnesota, TR 02-016.